Proximity Drawings of Binary Trees in Polynomial Area

P.Penna[*] *P.Vocca*[†]

Abstract

In this paper, we study weak β–proximity drawings. All known algorithms that compute (weak) proximity drawings produce representations whose area increases exponentially with the number of vertices. Additionally, an exponential lower bound on the area of (weak) proximity drawings of general graph has been proved. We present the first algorithms that compute a polynomial area β–proximity drawing of binary and ternary trees. The algorithms run in linear time.

1 Introduction.

β–proximity drawings [8] have been deeply investigated because of their interesting graphical features (see, e.g. [7, 12, 11, 1, 4, 5, 6, 3, 10]). β–proximity was first introduced by Kirkpatrick and Radke [8, 13] as a generalization of *Gabriel proximity* ([7, 12]) and Relative Neighborhood Graph ([14, 15]). A β–proximity drawing is a straight–line drawing where two vertices u and v are adjacent if and only if the region of the plane defined by the intersection of two disks, whose radii and centers depend on the parameter β, does not contain any other vertex except for u and v. This region is generally referred as *β–region of influence* or *β–proximity region* and the formal definition will be given in the sequel. In particular, a Gabriel drawing is a β–proximity drawing where the proximity region is the disk having as antipodal points u and v.

In [2], *weak β–proximity drawings* were first introduced. A weak β–proximity drawing relaxes the requirement of "classical" β–proximity drawings, allowing the β–region of non-adjacent vertices to be empty.

All known algorithms for (weak) β–proximity drawings produce representations whose area increases exponentially with the number of vertices, even when binary trees are considered [12, 11, 9, 4]. Additionally, for general graphs an exponential area lower bound exists [10]. Thus, the problem of constructing proximity drawings of graphs that have small area is considered a very challenging one by several authors (see [4, 5, 12]).

In this paper we describe a linear time algorithm for drawing binary trees with n vertices which produces a weak β–drawing that requires $O(n^2)$–area, for any $0 \leq \beta < \infty$. Additionally, we extend the result to ternary trees showing that they admit $O(n^2)$–area weak Gabriel drawing.

[*]Dipartimento di Scienze dell'Informazione, Università di Roma "La Sapienza",Via Salaria, 113, I–00198 Rome, Italy. E-mail: `penna@mat.uniroma2.it`.

[†]Dipartimento di Matematica, Università di Roma "Tor Vergata",Via della Ricerca Scientifica, I–00133 Rome, Italy. E-mail: `vocca@mat.uniroma2.it`.

2 Preliminaries and Notation.

A *layer* L_i is an horizontal line containing the points whose y–coordinate is Y_i, where Y_i is a positive integer. We assume that for any $i \geq 1$, $Y_{i+1} \geq Y_i$. A *layered drawing* in a straight line drawing such that each vertex is placed on a layer. In this definition we relax the assumption of classical definition of layered drawing that edges do not connect vertices on the same layer. Moreover, we allow layers not to be equally spaced. The *height*, *width* and the *area* of a drawing are the height, the width, and the area of the smallest isothetic rectangle bounding the drawing, respectively. Let a and b be two points whose distance is $d(a,b)$. We denote with $R[a,b,\beta]$ the *β–region of influence of a and b*. For any $0 < \beta < 1$, $R[a,b,\beta]$ is the intersection of the two closed disks of radius $d(a,b)/(2\beta)$ passing through both a and b. For any $1 \leq \beta < \infty$, $R[a,b,\beta]$ is the intersection of the two closed disks of radius $\beta d(a,b)$ and centered at the points $(1-\beta/2)a + (\beta/2)b$ and $(\beta/2)a + (1-\beta/2)b$. A drawing of a graph G is a *weak β–drawing* if for any pair of adjacent vertices a and b, the proximity region $R[a,b,\beta]$ does not contain any other vertex of the drawing. For $\beta = 1$, β–drawings are also known as *Gabriel* drawings. We denote $\alpha(\beta) = \inf\{\angle abc \mid c \in R[a,b,\beta]\} = 2\arcsin\sqrt{1/2\beta}$.

3 The Algorithm.

In this section, we describe the algorithm which produces a $O(n^2)$–area β–proximity drawing of binary trees. In Sect. 5, we modify the algorithm to produce a polynomial area Gabriel drawing of ternary trees.

Our algorithm takes two steps: (1) it constructs an *hv*–drawing Δ of tree t; (2) it vertically enlarges Δ to obtain a weak β–drawing Δ_β of t.

Drawing Δ satisfies the following two invariants:

Invariant 1 Vertical edges are one unit long.

Invariant 2 The width is at most n.

Drawing Δ_β is obtained from Δ by increasing the distance between consecutive layers.

Step 1

Drawing Δ can be recursively constructed as shown in Fig. 1. More formally, we denote with $\Delta_1 \ominus \Delta_2$ the drawing obtained by combining drawings Δ_1 and Δ_2 as follows: Δ_1 is translated to the bottom by one unit and Δ_2 is translated to the right by as many grid points as the width of Δ_1. The drawing Δ is constructed in linear time by algorithm `hv-draw` in Fig. 2. An example of *hv*–drawing is depicted in Fig. 4(b). Notice that, in general, an *hv*–drawing is not a proximity drawing.

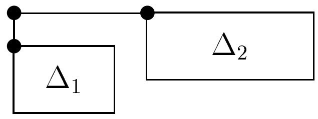

Figure 1: The drawing $\Delta_1 \ominus \Delta_2$.

```
algorithm hv-draw(t)
h ← height of t
r ← root of t
if h = 1 then
    draw r at (1,1)
    Δ ← drawing of r
else begin
    t1 ← smaller immediate subtree of t
    t2 ← larger immediate subtree of t
    Δ1=hv-draw(t1)
    Δ2=hv-draw(t2)
    Δ = Δ1 ⊖ Δ2
    end
return (Δ)
end
```

Figure 2: Algorithm `hv-draw`.

Step 2

Let us denote with l^i_{max} the maximum length of any horizontal edge of Δ on layer L_i (see the example of Fig. 4(b)). Algorithm `enlarge` in Fig. 3 computes Δ_β from Δ by spacing out each consecutive layers of a value proportional to $\delta(\beta) = \frac{1}{2 \tan \alpha(\beta)/2}$. We denote by x_v, y_v, x'_v, and y'_v the x– and y–coordinates of a vertex v in drawing Δ and Δ_β, respectively. By applying algorithm `enlarge` to the drawing Δ of Fig. 4(b) with $\beta = 2$ we obtain the proximity drawing of Fig. 4(c).

```
algorithm enlarge(Δ,β)
δ(β) ← 1 / (2 tan α(β)/2)
Y1 ← 1
for i = 1 to h do begin
    δi(β) ← ⌊δ(β) l^i_max⌋ + 1
    Yi ← Y(i−1) + δi(β)
    for each vertex v on layer Li do
        x'v ← xv; y'v ← Yi
    end
draw the edges;
return (Δβ)
end
```

Figure 3: Algorithm `enlarge`.

4 Proof of Correctness.

In this section we prove that, for any β, algorithm `enlarge` returns a weak β–drawing of $O(n^2)$–area.

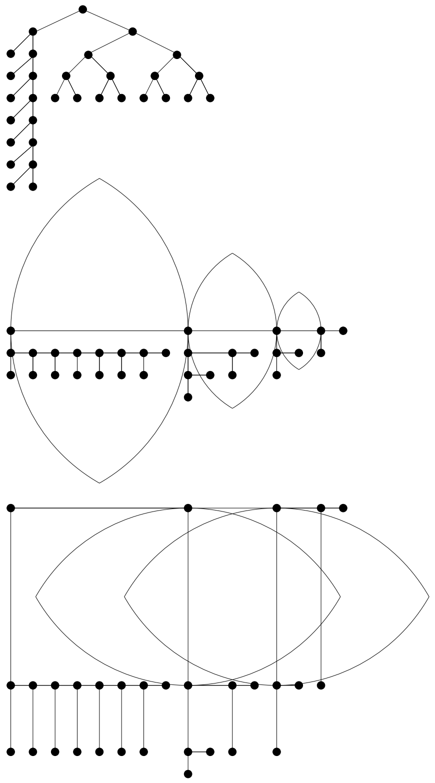

Figure 4: Binary Trees: an example. (a) The tree given in input. (b) The first step (hv–drawing). (c) The β–proximity drawing with $\beta = 2$.

We denote with h and n the height and the number of vertices of the tree, respectively. Moreover, n_1 and n_2 are the number of nodes of the smaller and larger immediate subtrees. We first need the following lemma.

Lemma 4.1 *For any* $1 \leq i \leq h$, $l^i_{max} \leq n/2^i$.

Proof. The proof proceeds by induction on n. *Base Step* For $n = 1$, $l^i_{max} = 1$. *Induction Step* From algorithm `hv-draw`, from the fact that $n_1 \leq n/2$ and $n_2 \leq n$, we have:
for $i = 1$

$$l^1_{max} = \max\{n_1, n_2/2\} \leq n/2,$$

and for $2 \leq i \leq h$

$$l^i_{max} = \max\{n_1/2^{i-1}, n_2/2^i\} \leq n/2^i.$$

□

Lemma 4.2 *For any* $0 \leq \beta < \infty$, Δ_β *is a weak* β*–drawing.*

Proof. We have to prove that for any edge (a, b), $R[a, b, \beta]$ does not contain any other vertex. We distinguish two cases:

Horizontal edges. Let c be any point on layer L_{i+1}. Because of the definition of $\delta_i(\beta)$, $\angle acb < \alpha(\beta)$. Similarly, since $\delta_i(\beta) < \delta_{i-1}(\beta)$, then $\angle ac'b < \alpha(\beta)$ for any c' on L_{i-1}. Thus, from the definition of $\alpha(\beta)$ it follows that $R[a, b, \beta]$ neither intersect L_{i+1} nor L_{i-1}.

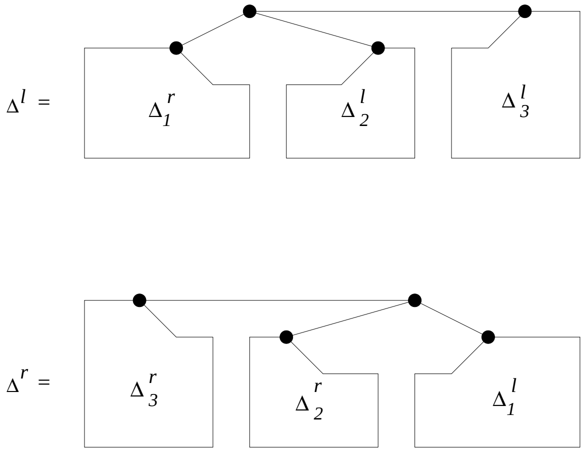

Figure 5: Drawings of ternary trees: first step.

Vertical edges. Let a be on layer L_i and b on layer L_{i-1}. From **Invariant 1** and considering that Δ_β is a layered drawing, $R[a, b, \beta]$ intersects a on layer L_i and b on L_{i-1}, only.

□

Lemma 4.3 *The area of Δ_β is at most $n^2(\delta(\beta) + 1)$.*

Proof. From the definition of $\delta_i(\beta)$ and from Lemma 4.1 it follows that the height of Δ_β is at most

$$\sum_{i=1}^{h} \delta_i(\beta) \leq \sum_{i=1}^{h} \left(\delta(\beta) l_{max}^i + 1\right) \leq \sum_{i=1}^{h} \left(\frac{n\delta(\beta)}{2^{i+1}} + 1\right) < n\delta(\beta) + n.$$

Thus from **Invariant 1**, the area of Δ_β is at most $n^2(\delta(\beta) + 1)$. □

Hence, from Lemmas 4.2 and 4.3, we can state the following result.

Theorem 4.4 *For any $0 \leq \beta < \infty$ and for any binary tree t with n nodes, an $O(n^2)$–area weak β–drawing of t exists.*

5 Ternary Trees

In this section we extend the previous result to ternary trees. Unfortunately, this is not a complete extension as we can manage the class of β–proximity drawings for $\beta \leq 1$, only, which properly includes Gabriel drawings.

We use a two–steps technique similar to the one above described for binary trees. In particular, the first step, shown in Fig. 5, is suitably changed, while the second is substantially the same.

Theorem 5.1 *Any ternary tree admits $O(n^2)$–area weak Gabriel drawing.*

An example of the Gabriel drawing of a ternary tree obtained from our algorithm is shown in Fig. 6. It is easy to see that for $\beta > 1$ the above construction does not guarantee the proximity regions of diagonal edges to be empty.

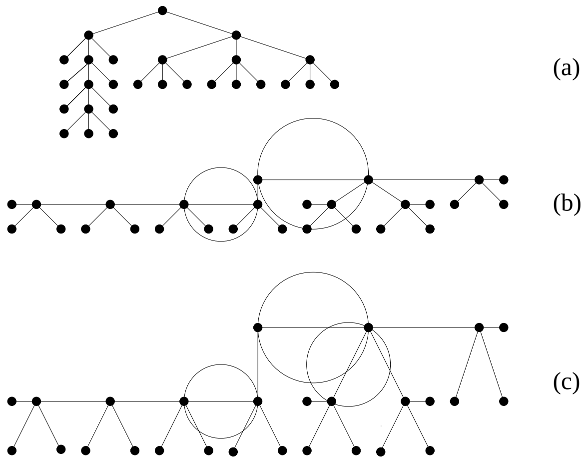

Figure 6: Ternary trees: an example. (a) The tree given in input. (b) The first step. (c) The proximity drawing.

6 Conclusions and Open Problems

In this paper we have shown that binary and ternary trees admit polynomial area β–proximity and Gabriel drawings, respectively. Moreover, the algorithms we presented both take linear time. Several problems are open:

1. Prove a lower bound for binary and ternary trees;
2. Try to extend the results to trees of degree greater than three;
3. Investigate other classes of graphs that admit β–drawings (e.g. outerplanar graphs);
4. Consider strong proximity drawings.

Acknowledgments. The authors warmly thanks Prof. Roberto Tamassia and Ing. Giuseppe Liotta for their suggestions and helpful discussions. The problem tackled in this paper was raised and a preliminary study was done while the second author was visiting the Center for Computational Geometry at Brown University.

References

[1] G. Di Battista, W. Lenhart, and G. Liotta. Proximity drawability : a survey. In *Proc. Graph Drawing '94*, Lecture Notes in Computer Science, pages 328–339. Springer Verlag, 1994.

[2] G. Di Battista, G. Liotta, and S.H. Whitesides. The strenght of weak proximity. In F. J. Brandenburg, editor, *Graph Drawing (Proc. GD '95)*, volume 1027 of *Lecture Notes in Computer Science*, pages 178–189. Springer–Verlag, 1996.

[3] P. Bose, G. Di Battista, W. Lenhart, and G. Liotta. Proximity constraints and representable trees. In *Proc. Graph Drawing '94*, LNCS, pages 340–351. Springer–Verlag, 1994.

[4] P. Bose, W. Lenhart, and G. Liotta. Characterizing proximity trees. *Algorithmica*, 16:83–110, 1996.

[5] P. Eades and S. Whitesides. The realization problem for euclidean minimum spanning tree is NP-hard. In *Proc. ACM Symp. on Computational Geometry*, pages 49–56, 1994.

[6] H. ElGindy, G. Liotta, A. Lubiw, H. Meijer, and S. H. Whitesides. Recognizing rectangle of influence drawable graphs. In *Proc. Graph Drawing '94*, number LNCS, pages 352–363. Springer–Verlag, 1994.

[7] K. R. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18:259–278, 1969.

[8] D.G. Kirkpatrick and J.D. Radke. A framework for computational morphology. In G.T.Toussaint, editor, *Computational Geometry*, pages 217–248, Amsterdam, Netherlands, 1985. North–Holland.

[9] W. Lenhart and G. Liotta. Proximity drawings of outerplanar graphs. In Stephen North, editor, *Graph Drawing (Proc. GD '96)*, volume 1190 of *Lecture Notes in Computer Science*, pages 286–302, 1997.

[10] G. Liotta, R. Tamassia, J. G. Tollis, and P. Vocca. Area requirement of Gabriel drawings. In Giancarlo Bongiovanni, Daniel Pierre Bovet, and Giuseppe Di Battista, editors, *Proc. CIAC'97*, volume 1203 of *Lecture Notes in Computer Science*, pages 135–146. Spriger–Verlag, 1997.

[11] A. Lubiw and N. Sleumer. All maximal outerplanar graphs are relative neighborhood graphs. In *Proc. CCCG'93*, pages 198–203, 1993.

[12] D. W. Matula and R. R. Sokal. Properties of Gabriel graphs relevant to geographic variation research and clustering of points in the plane. *Geogr. Anal.*, 12:205–222, 1980.

[13] J.D. Radke. On the shape of a set of points. In G.T. Toussaint, editor, *Computational Morphology*, pages 105–136, Amsterdam, The Netherlands, 1988. North–Holland.

[14] G.T. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern recognition*, 12:261–268, 1980.

[15] R. B. Urquhart. Some properties of the planar euclidean relative newighborhood graph. *Pattern recognition Letters*, 1:317–332, 1983.